COMPILADOR ZX-SPECTRUM (16K/48K). Copyright: WYE VALLEY SOFTWARE

Producido y distribuído en exclusiva en España por: VENTAMATIC Micro-Informática - Chalet "Capvespre" - Avda. de Rhode, nº 253 - Apartado de Correos nº 168 - Tel.: (972) 257 985 / 255 616 - ROSES (Girona).

Este Compilador convertirá sus programas en BASIC a programas en código máquina. Hablando literalmente, el código máquina es "la lengua nativa" del ZX-SPECTRUM, por lo que permite que los programas se ejecuten mucho más rápidamente. El incremento de velocidad puede ser de hasta varios cientos de veces más rápido que el programa normal en BASIC.

Este Compilador se suministra en dos versiones en una sola cinta, una para el ZX-SPECTRUM de 16K y otra para el ZX-SPECTRUM de 48K. De ahora en adelante, en estas instrucciones, las direcciones para la versión de 48K se indicarán entre paréntesis () al lado de las direcciones correspondientes para la versión de 16K.

USO

Cargue el Compilador en el ordenador con LOAD "". Una vez cargado pulse cualquier tecla para borrarlo el programa introductorio: el Compilador estará localizado en la parte alta de memoria a salvo de NEW.
Escriba su programa en BASIC (o cárguelo del cassette) y ejecútelo para comprobar que no tenga errores. Cuando todo esté totalmente correcto escriba RAND USR 27ØØØ (6ØØØØ) y pulse ENTER para compilar su programa. Verá aparecer algunos mensajes en la pantalla.

Estos mensajes tienen los siguientes significados:

1. "START ADRESS" (Dirección Inicial).
Indica la dirección inicial del código máquina que se está produciendo en la compilación (esta dirección es siempre un byte menos que RAMTOP).

2. "END ADRESS" (Dirección Final).
Esta dirección se va actualizando durante la compilación e indica la última dirección del código máquina que se ha creado.

3. "ARRAY END ADRESS" (Dirección Final de la Tabla).
Este mensaje solo aparece cuando se usa la TABLA permitida en el programa. La tabla se almacena directamente detrás del código máquina. Este mensaje indica la última dirección de la tabla tras haber completado la compilación.

4. "FIRST PASS" (Primer Paso), "SECOND PASS" (Segundo Paso).
Para efectuar una compilación completa, el Compilador debe codificar el programa dos veces. Este mensaje indica en qué etapa de compilación se encuentra.

5. "ERROR", "NO ERRORS".
Si una vez compilado el programa por completo aparece el mensaje "NO ERRORS", entonces es que ya dispone del código máquina correcto. Si en cualquier etapa aparece el mensaje "ERROR" en la pantalla, es que el Compilador ha encontrado un fallo en su programa BASIC.
En este caso el Compilador mostrará la línea incorrecta y pondrá un interrogante (?) después del error. Cambiará también la "línea actual" a la que contenga el error, y por lo tanto bastará con pulsar EDIT (SHIFT 1) para editarla y corregirla. Después habrá que re-iniciar la compilación desde el principio.

EJECUCION DE PROGRAMAS COMPILADOS

Tome nota de la dirección inicial ("START ADRESS") que se indica durante la compilación y luego escriba RAND USR (START ADRESS) para ejecutar el código compilado.

DONDE SE UBICA EL CODIGO MAQUINA

Se puede asignar la dirección a partir de la cual se desea ubicar el código máquina creado durante la compilación mediante CLEAR N (donde N es una unidad menos que la dirección inicial requerida).
Al cargar el Compilador en el ZX-SPECTRUM, RAMTOP se ajusta automáticamente a 26ØØØ (4ØØØØ).
Mediante este sistema se pueden compilar varios programas, colocándolos uno detrás de otro, o simplemente en áreas de memoria distintas.

MEMORIA

La memoria disponible puede agotarse de varias maneras.

1. Durante la compilación se crea una tabla de direcciones de los NUMEROS DE LINEA y puede ocurrir un error "OUT OF MEMORY" (Fuera de memoria) en el caso de que se utilicen demasiados números de línea.

2. Si el código máquina creado o la TABLA intentan sobre-escribir el compilador también ocurrirá un error "OUT OF MEMORY".

VARIABLES Y NUMEROS

Hay 52 variables permitidas:
- 26 letras Mayúsculas.
- 26 letras Minúsculas.

Se permite también dimensionar una TABLA UNI-DIMENSIONAL A() que puede tener hasta 255 elementos.

Los números se almacenan únicamente en forma entera, en el rango de -32767 a 32767.

Algunas funciones tomarán los números a almacenar en el rango de Ø a 65536. Por ejemplo: POKE, PEEK, USR, etc.

Todas las operaciones pueden escribirse de la manera normal, pero deben encerrarse entre paréntesis ().

COMANDOS QUE PUEDE MANEJAR EL COMPILADOR

El Compilador puede manejar un amplio conjunto de los comandos normales BASIC. Incluso hay algunos comandos muy avanzados que el Compilador puede manejar, y el BASIC normal no.
He aquí una lista de los comandos permitidos:

CLAVES

a	representa una variable de una letra.
c	representa una secuencia de atributos de color, separados por ";" ó ",". Un atributo de color tiene la forma de PAPER, INK, FLASH, BRIGHT, INVERSE o OVER.
s	representa una secuencia de sentencias separadas por dos puntos ":".
x,y,z	representan expresiones numéricas.
e	representa una expresión.
n	representa un número.

COMANDOS

BEEP x,y	Esta sentencia se ha cambiado de la normal en BASIC para hacerla más versátil. X e Y pueden ser cualquier número entero. X da la duración (en realidad es el número de ciclos de onda cuadrada sacados y debe ser calculado para cada frecuencia). Y es el retardo entre cada ciclo de onda cuadrada sacado, por lo tanto decrementando Y aumenta la frecuencia. Esto puede parecer complicado, pero permite la generación de sonidos extremadamente complejos.
BORDER e	Igual que en BASIC normal.
BRIGHT e	Igual que en BASIC normal.
CIRCLE c;x,y,z	Igual que en BASIC normal.
CLEAR	Pone todas las variables a cero, pero deja intacta la tabla.
CLS	Igual que en BASIC normal.
COPY	Igual que en BASIC normal.
DATA e1,e2,e3	Igual que en BASIC normal.
DIM A(n)	Dimensiona una tabla con el nombre A().
DRAW c;x,y	Igual que en BASIC normal.
FLASH e	Igual que en BASIC normal.
FOR a=x TO y	Igual que en BASIC normal.
FOR a=x TO y STEP z	Igual que en BASIC normal.
GOSUB n	Igual que en BASIC normal.
GO TO n	Igual que en BASIC normal.
IF x THEN s	Igual que en BASIC normal (>,=,< se pueden utilizar).

INK e	Igual que en BASIC normal.
INPUT	Igual que en BASIC normal (este es exactamente el mismo que en el BASIC normal del ZX-SPECTRUM ya que pueden incluirse textos como en un PRINT y pueden entrarse varias variables). Normalmente, los caracteres tecleados aparecen en la parte de abajo de la pantalla, pero si se desea, cambiando el connido de una dirección de memoria antes de compilar, pueden aparecer en la parte de arriba de la pantalla. Esta dirección es 2793Ø (6Ø93Ø). Su contenido debe ser Ø para un INPUT normal y 2 para un INPUT especial.
INVERSE e	Igual que en BASIC normal.
LET a=e	Igual que en BASIC normal.
LOAD f CODE	Igual que en BASIC normal.
LPRINT	Igual que en BASIC normal.
NEXT a	Igual que en BASIC normal.
OUT x,y	Igual que en BASIC normal.
OVER e	Igual que en BASIC normal.
PAPER e	Igual que en BASIC normal.
PAUSE e	Igual que en BASIC normal.
PLOT c;m,n	Igual que en BASIC normal.
POKE m,n	Igual que en BASIC normal.
PRINT	Igual que en BASIC normal.
RANDOMIZE	Igual que en BASIC normal.
READ a	Igual que en BASIC normal.
REM	Esta sentencia puede estar seguida de una letra sóla y algunos parámetros, siendo cada letra un comando que no está disponible normalmente en el BASIC del ZX-SPECTRUM. Las letras que son aceptadas por el Compilador como comandos válidos detrás de un REM se listan más adelante. Si el primer caracter después del REM no es una de estas letras, el REM se trata de la manera normal.
RESTORE n	Igual que en BASIC normal.
RETURN	Igual que en BASIC normal.
SAVE f CODE m,n	Igual que en BASIC normal.
STOP	Provoca la devolución del control al BASIC con un mensaje de STOP.

COMANDOS EXTRA (DESPUES DE SENTENCIAS REM)

A,e	Asigna el color del atributo e.
B	Comprueba si la tecla BREAK está pulsada.
E,e	Provoca un error número e+1.
F	Cambia la pantalla completa al atributo en curso, borrándola al mismo tiempo.
L,e	Desplaza la línea e 1 pixel hacia la izquierda.
R,e	Desplaza la línea e 1 pixel hacia la derecha.
N	Saca ruido aleatorio por el altavoz (explosión).
S,e,x,y	Gráficos Mini-Sprite: Este comando coloca el caracter número e en las coordenadas de PLOT x,y. Usando dos o tres de estas sentencias pueden obtenerse gráficos en movimiento muy espectaculares.

FUNCIONES

ATTR n,m
CHR$ n
CODE "carácter"
CODE INKEY$
CODE SCREEN$
IN n
PEEK n
POINT n,m
RND obtiene un número aleatorio entre Ø y 32767.
USR e + USR "caracter"

ALMACENAMIENTO DEL CODIGO MAQUINA

Para que el Compilador produzca código máquina eficientemente, debe haber un conjunto de rutinas en tiempo de ejecución a las que se accede desde el código. Los programas en código máquina no pueden funcionar sin estas rutinas, por lo que deben ser almacenadas junto con el código máquina.

El programa puede ser almacenado como sigue:
Para el ZX-SPECTRUM de 48K usar:
SAVE "nombre" CODE dirección inicial, 65536 - dirección inicial.

Para el ZX-SPECTRUM de 16K usar:
SAVE "nombre" CODE dirección inicial, 32767 - dirección inicial.

EJEMPLOS

1.

```
1Ø FOR a=4Ø TO 2ØØ STEP 5
2Ø FOR b=5Ø TO 6ØØ STEP a
3Ø BEEP 5Ø,b
4Ø NEXT b
5Ø NEXT a
```

2.

```
1Ø FOR A=1 TO 1ØØ
2Ø REM N
3Ø REM B
4Ø NEXT A
```

3.

```
1Ø FOR a=1 TO 128
15 LET b=(255-a)
2Ø REM S,87,a,a
21 REM S,88,b,a
3Ø REM B
31 BEEP 5,b
4Ø NEXT a
45 BORDER Ø:PAPER Ø:INK 7:REM F
5Ø REM N
55 BORDER 7:PAPER 7:INK Ø:CLS
6Ø REM N
```